UNIVERSITÀ DEGLI STUDI DI UDINE

hic sunt futura

ALPEN-ADRIA UNIVERSITÄT

KLAGENFURT I WIEN GRAZ

Multimedia Communication and Information Technology

Deep learning application for real-time ray tracing

Master Degree

Supervisors:

Prof. Lauro Snidaro

Prof. Bernhard Rinner

Graduating Student:

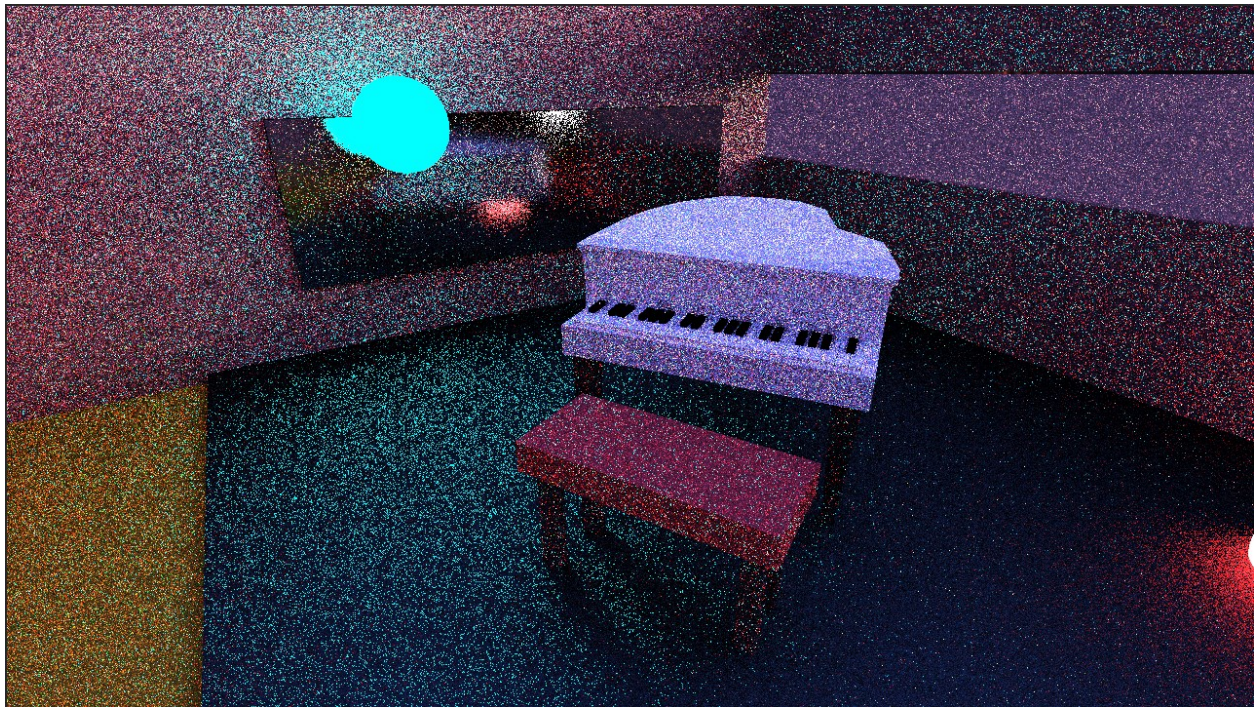
Michele Somero

UNIVERSITÀ DEGLI STUDI DI UDINE

UNIVERSITÄT KLAGENFURT

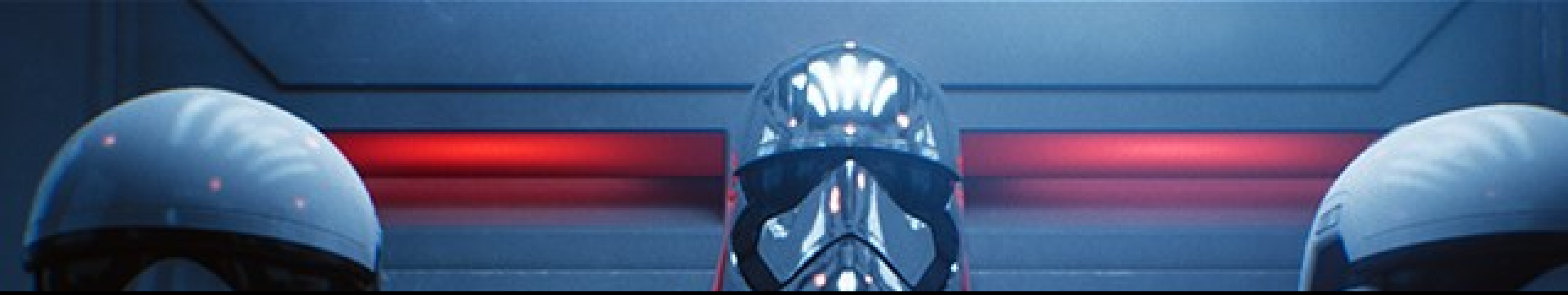
Thesis objectives

- The use of ray tracing in real-time applications
- Machine Learning approach for enhancing the ray tracing result



Real time application development using Unity:

- Ray tracing via compute shaders
- Frame denoising via Convolutional Neural Network



Real-time rendering
State of Art

Real-time rendering

- Application that requires interaction with the scene
  - Videogames
  - Simulators
  - Other real-time applications

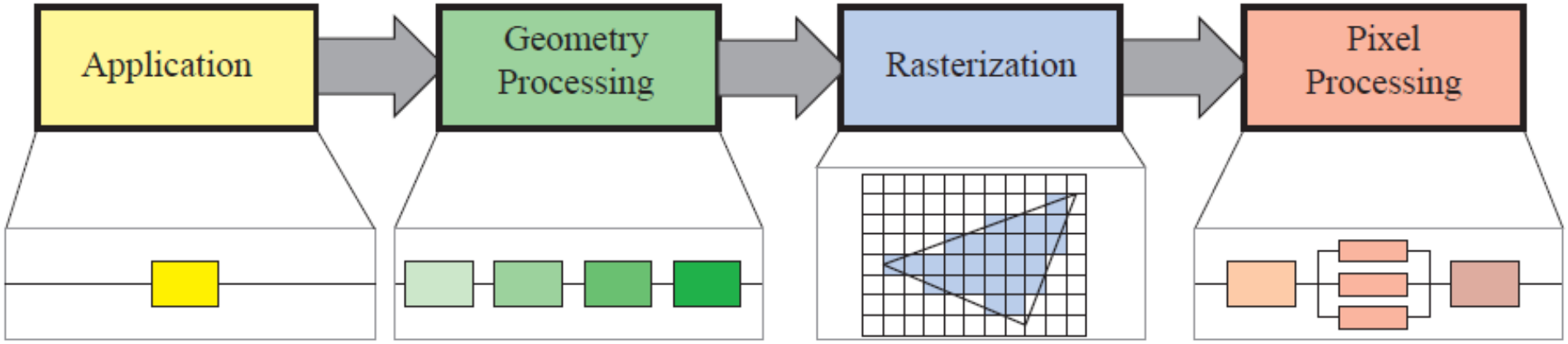
AR application for tourism

Rare's Sea of Thieves video game

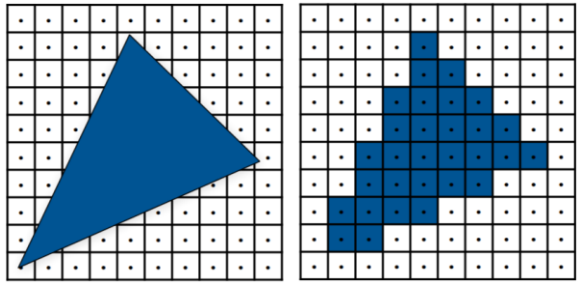
Microsoft Flight Simulator 2020

Rasterization

- All information of the mesh are converted into a series of pixel
  - Application: CPU computes changes in the scene
  - Geometry processing: GPU computes changes in geometry
  - Rasterization: match each pixel with a vertex or geometry
  - Pixel Processing: each fragment computes the color of the scene



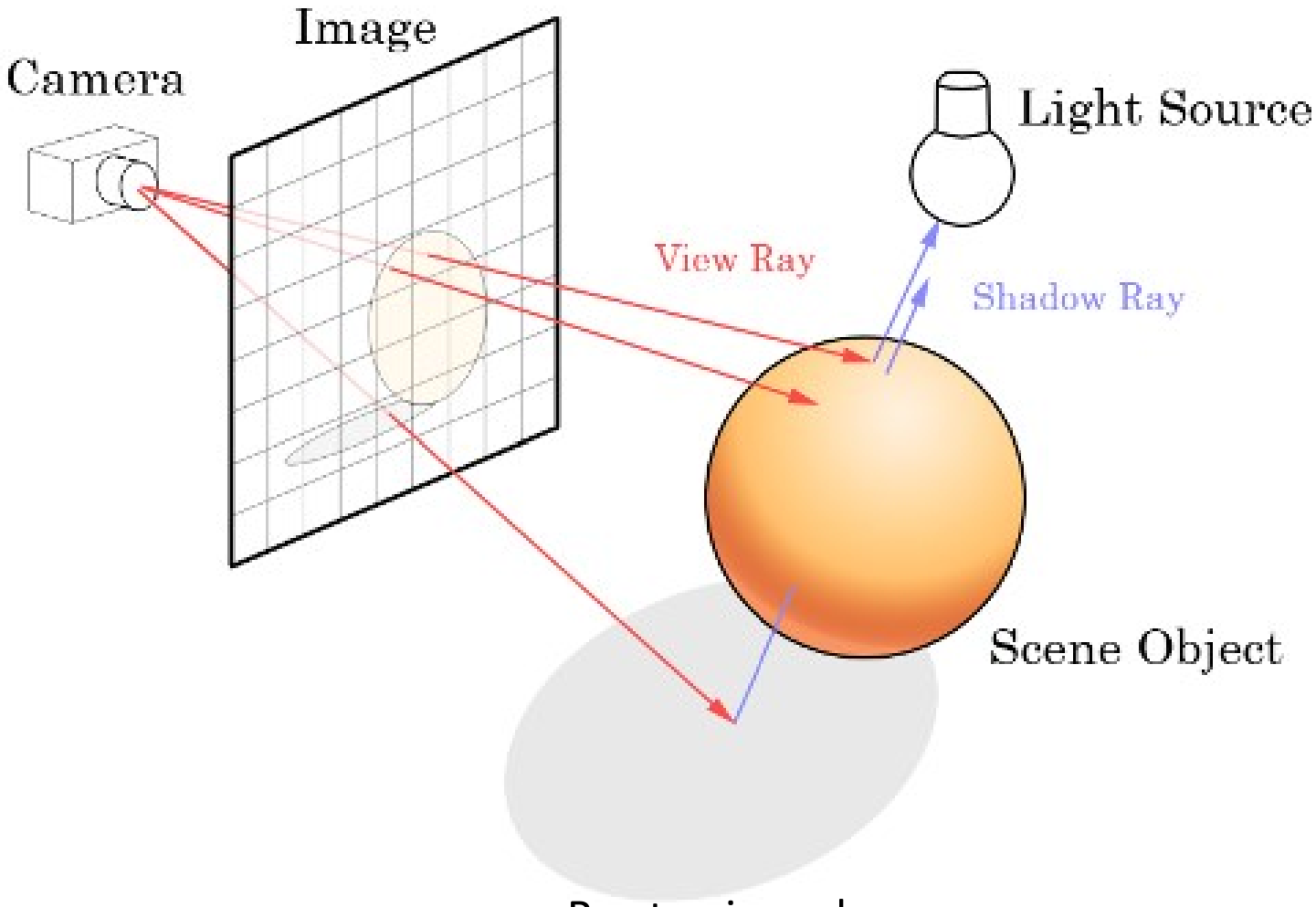
Rasterization pass:



Real-time rendering pipeline with rasterization

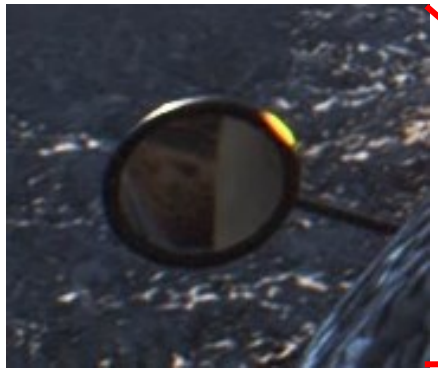
Ray tracing

- Per each pixel one or more ray are shoot in the scene to determinate the color and illumination of the scene itself



Ray tracing scheme

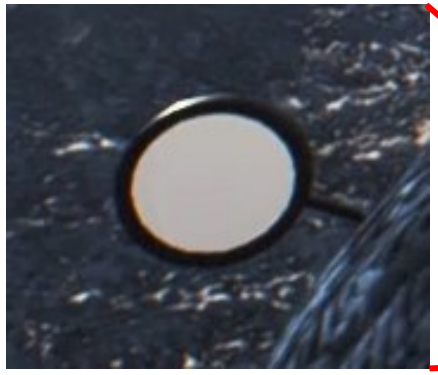
Ray tracing effects and problems



Nvidia RTX on

EA's Battlefield V with Nvidia RTX on

Nvidia RTX on

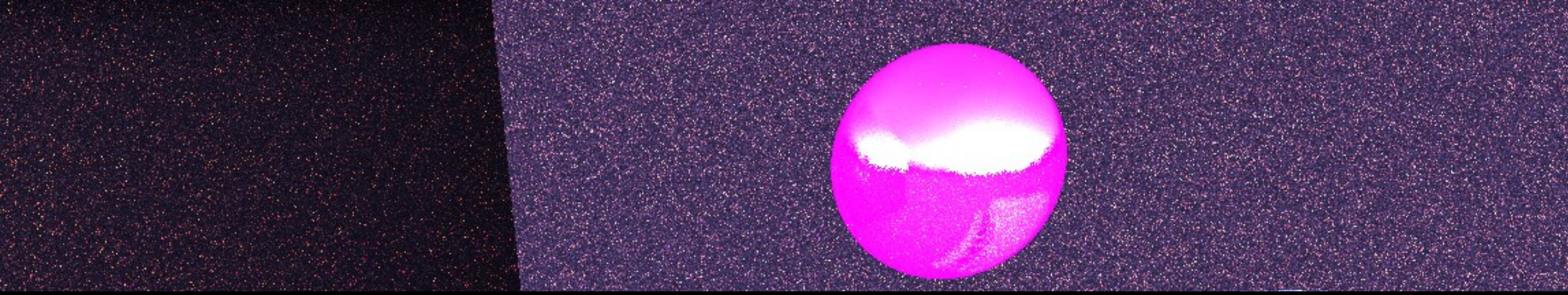


Nvidia RTX off

EA's Battlefield V with Nvidia RTX off

Nvidia RTX off

- High amount of pixel in a high-resolution image (1920 x 1080 px)
- Noisy image as output of a fast ray tracing (1spp)
- Demand of a high-end computer in order to ray trace in real time



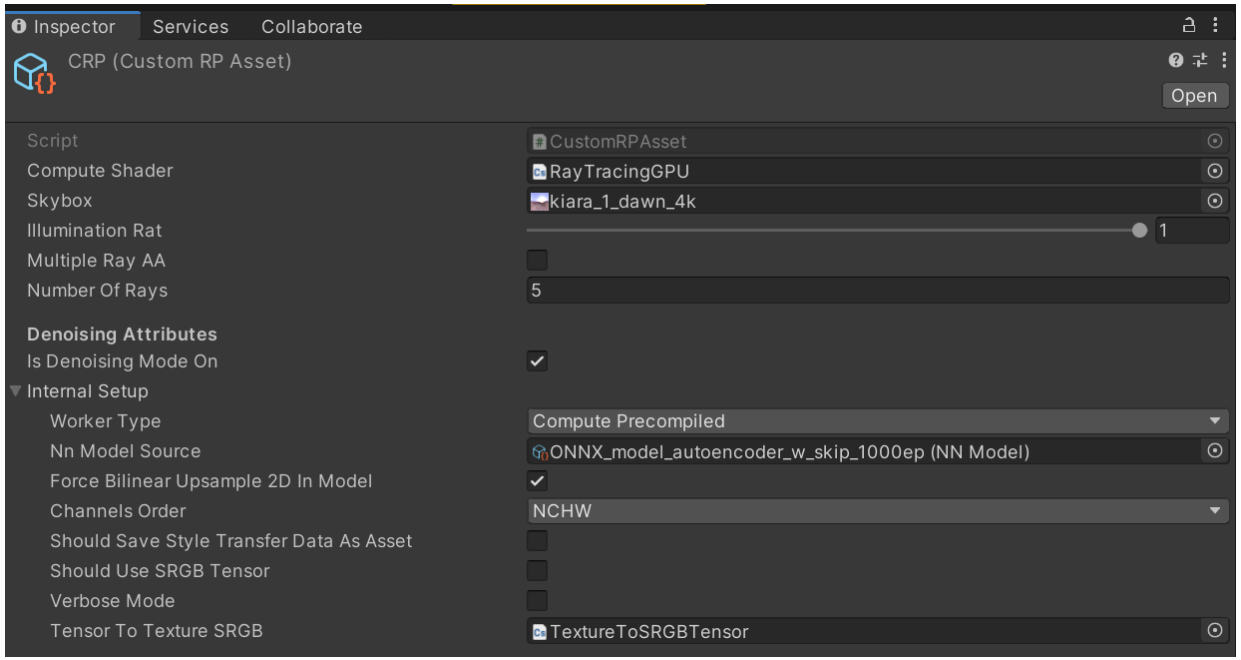
Real-time ray tracing project

UNIVERSITÀ DEGLI STUDI DI UDINE

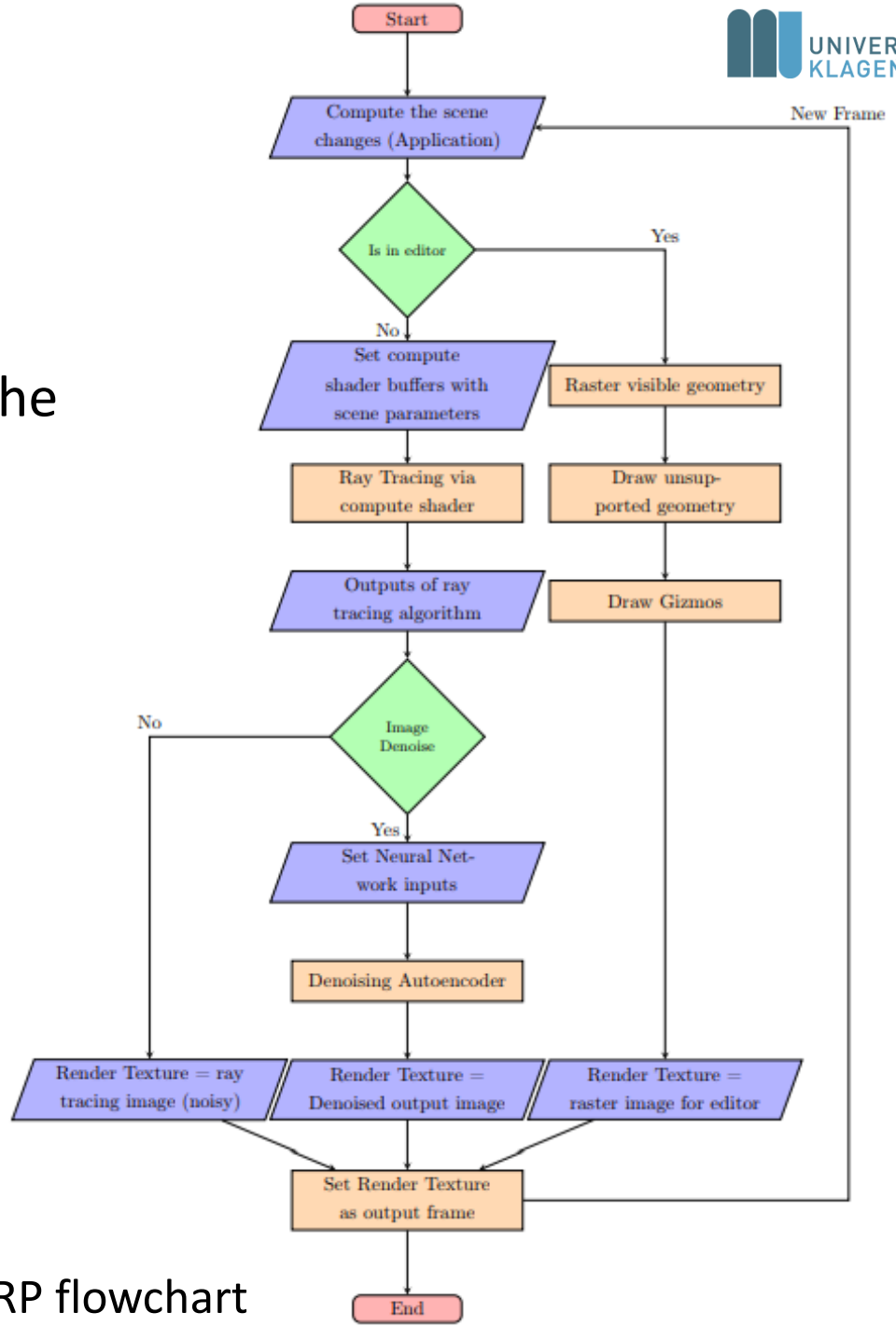
UNIVERSITÄT KLAGENFURT

Ray tracing using Compute Shaders and SRP

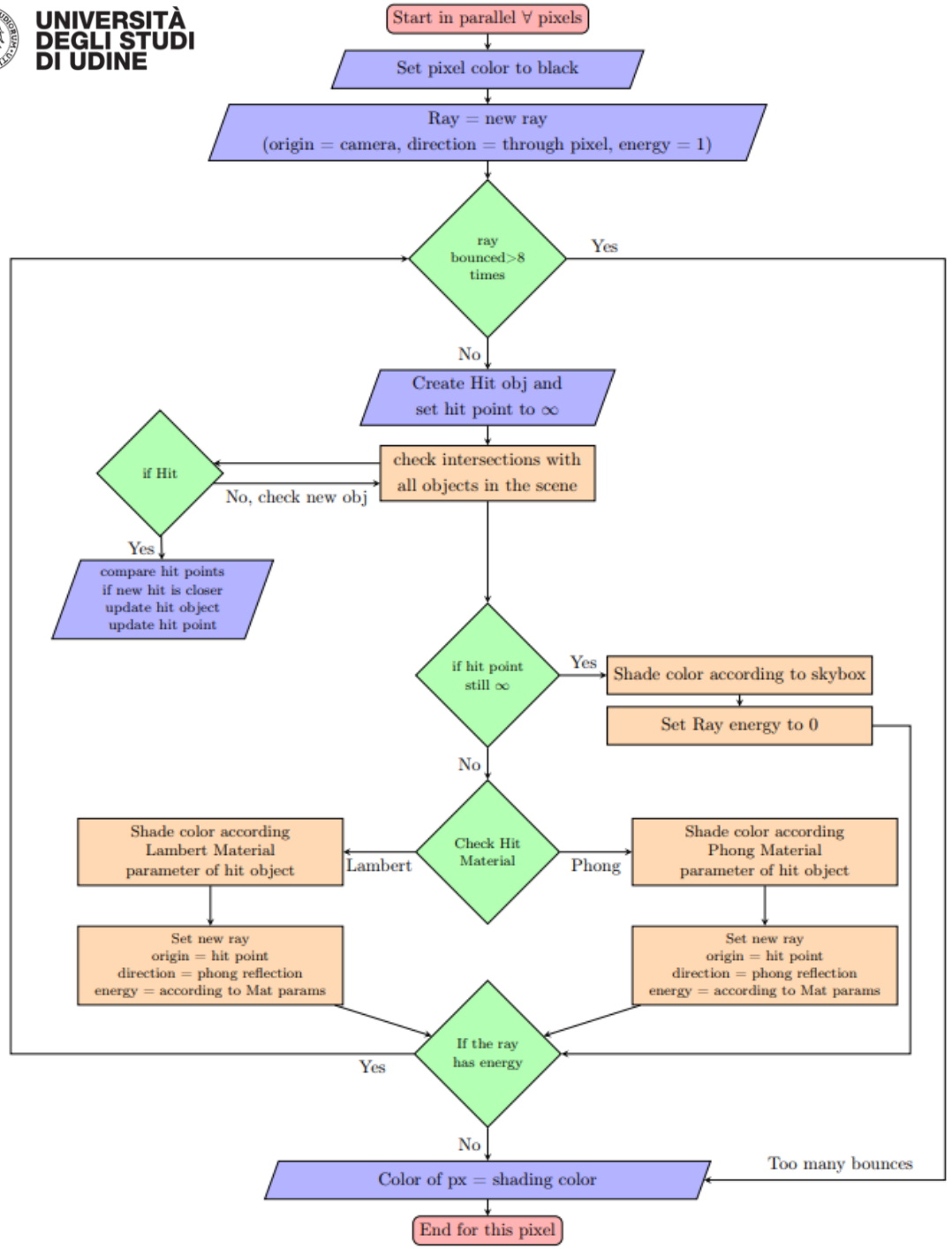
- Compute Shaders: code that run on the GPU outside the Pipeline
- SRP (Scriptable Rendering Pipeline): fully customable render pipeline in Unity



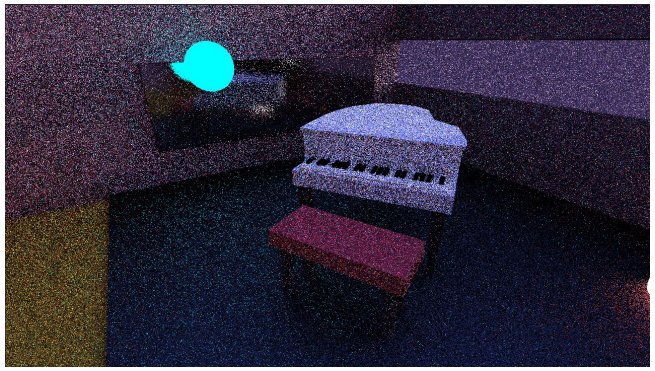
SRP object in Unity



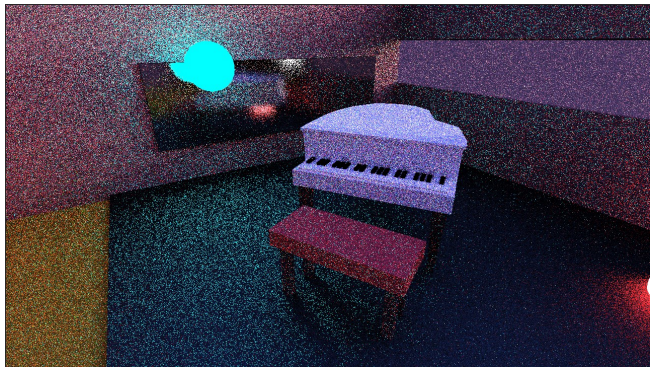
SRP flowchart



- Ray tracing
- Per each pixel, ray generation and soot
- Ray-objects intersection
- Ray params update
- Pixel color update



1 sample per pixel image with compute shader ray tracing

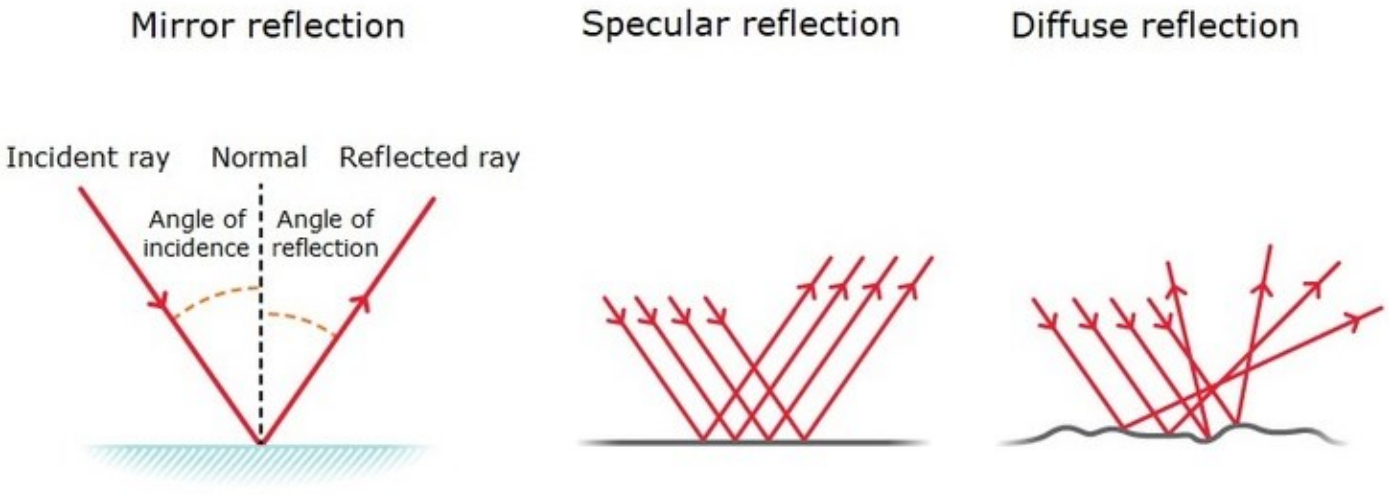


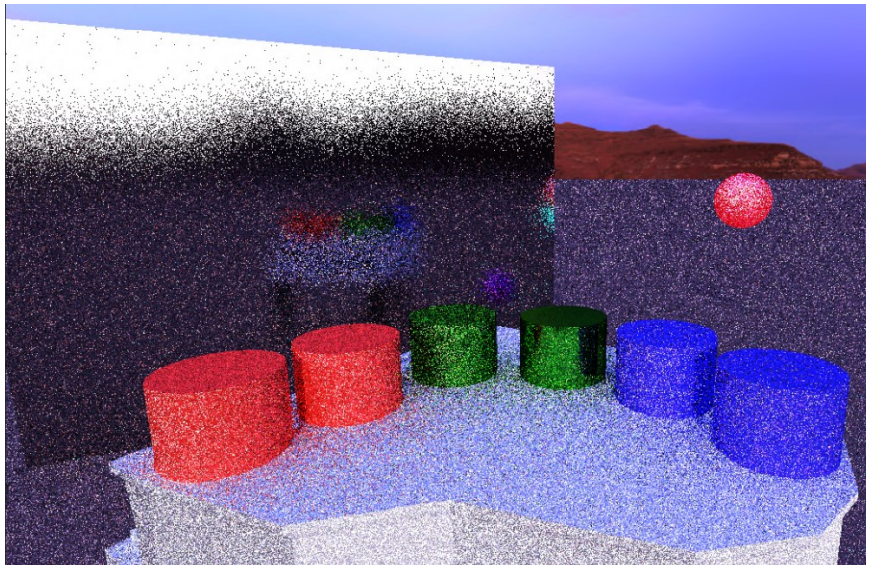
5 sample per pixel image with compute shader ray tracing

Materials and ray tracing

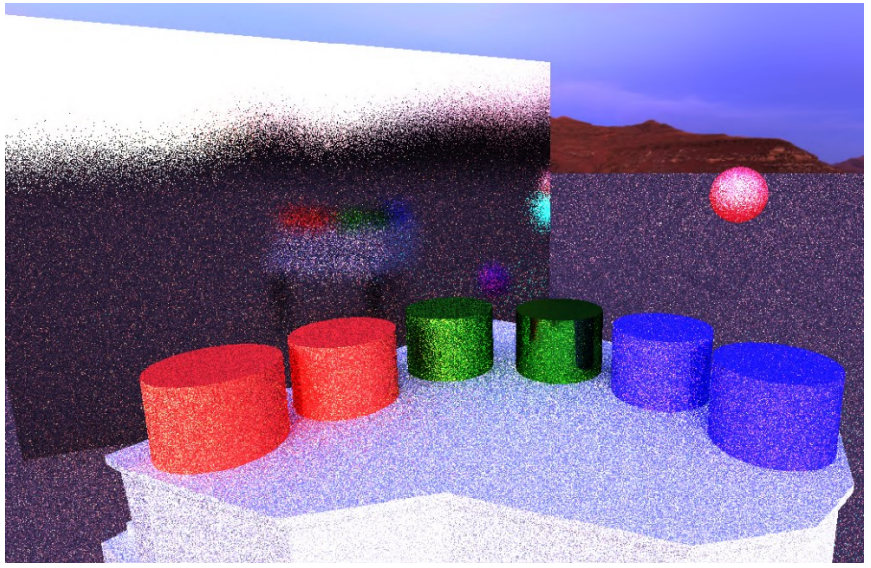
Different material types with different ray-material interactions

- Emissive
- Mirrors
- Lambertian
- Phong





1spp image materials behavior



5spp image materials behavior

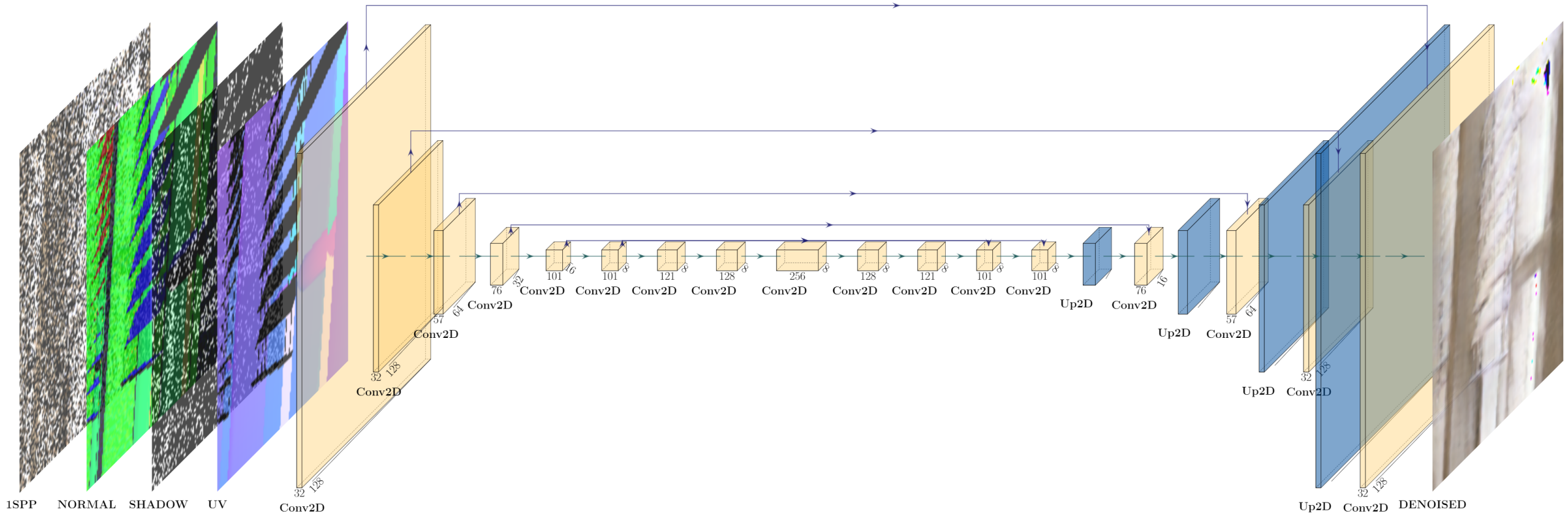
Machine Learning for image Denoising - project

UNIVERSITÀ DEGLI STUDI DI UDINE

UNIVERSITÄT KLAGENFURT

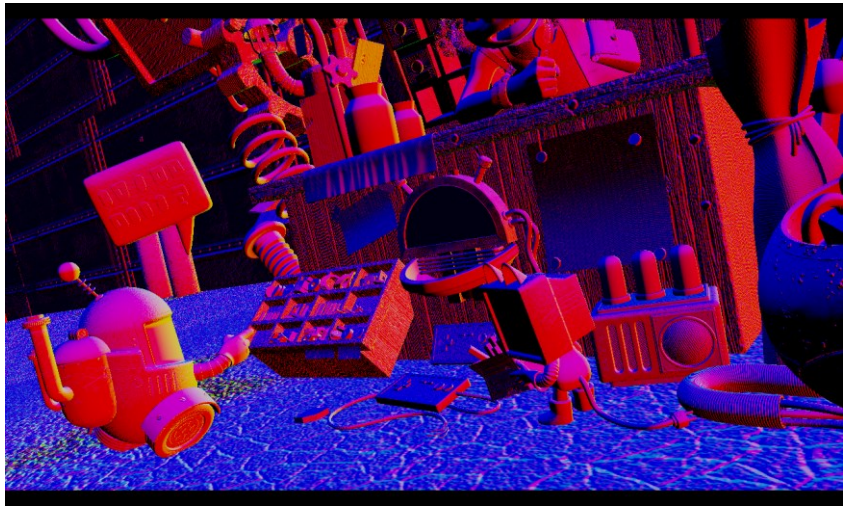
Convolutional Neural Network Denoiser

Architecture



Dataset

1 sample per pixel image



Normal map image

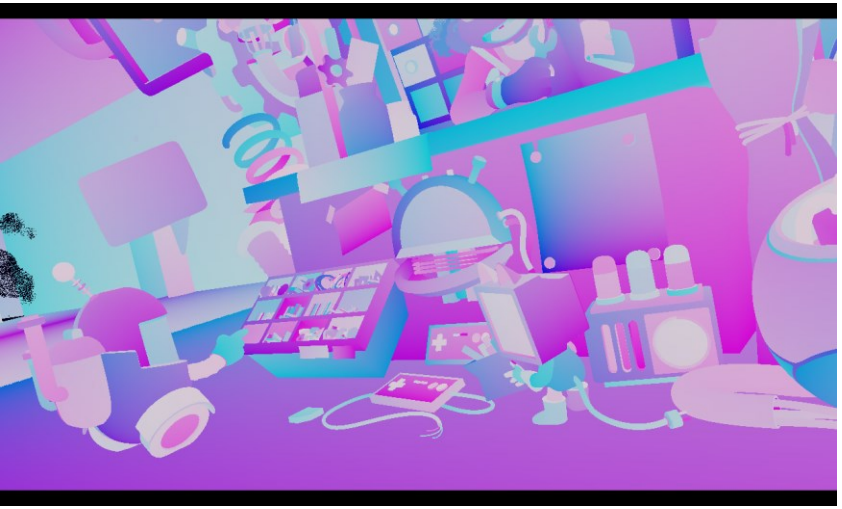
Database composed by 168 group of images of resolution *1280x720 px*

Each group composed by 5 images

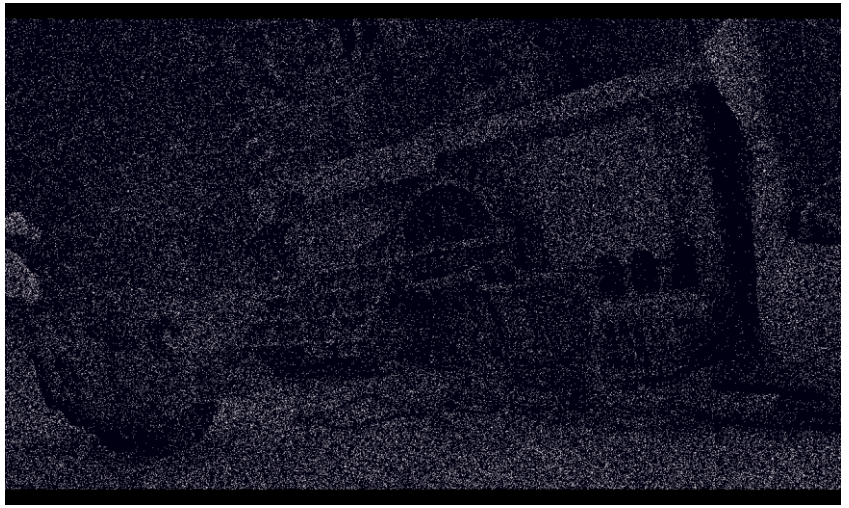
Training images have resolution *128x128 px* and are a crop of these image

Training subdivision:

- Trainset = 75%
- Testset = 19%
- Validationset = 6%



UV map image



Shadow image

Reference image (training only)

Evaluations and Conclusions

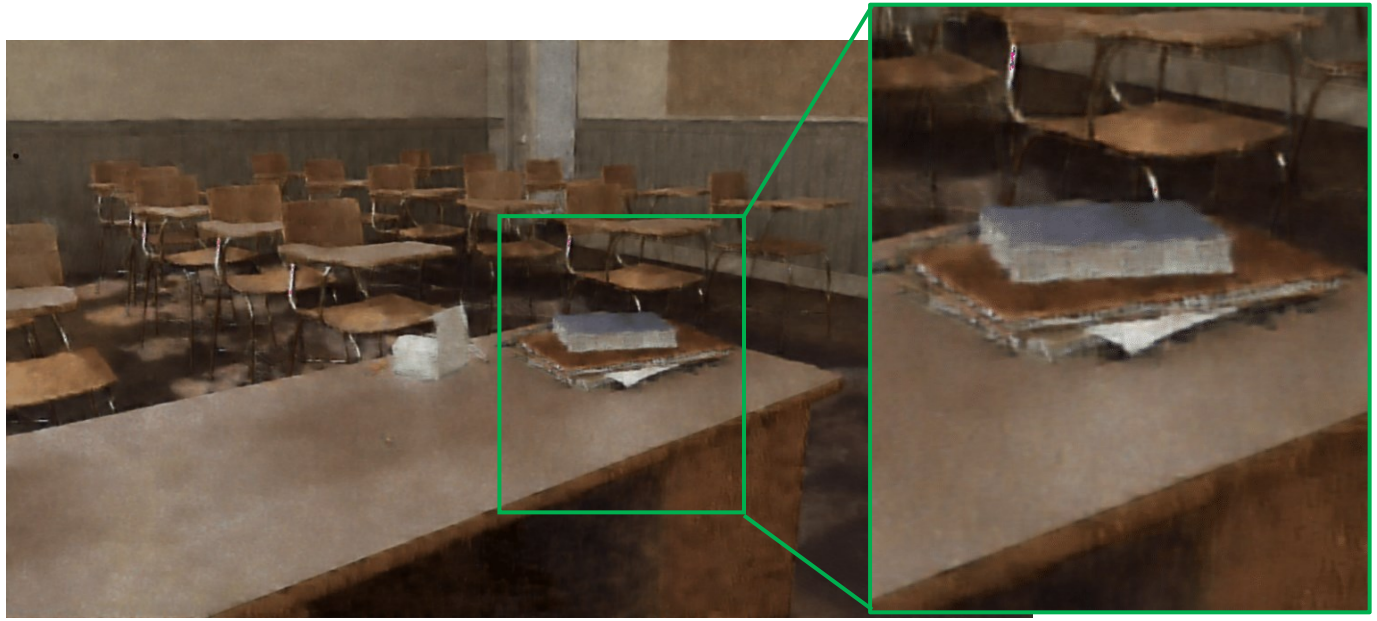
Denoiser evaluation

Denoiser trained with 600 epochs:

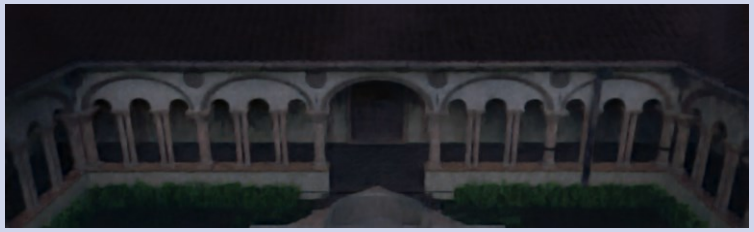
- Details More coherent
- Almost all noise reduction

Denoiser trained with 150 epochs:

- Details Problems
- Noise reduction not good



Denoiser evaluation

Image SPP	Set	Image denoised Example	Prediction time	Loss	Accuracy
1spp	Validation		53 ms	0,0276357395	0,5980369449
1spp	New		380 ms	0,0309227929	0,4627912939
5spp	New		51 ms	0,0269892520	0,5277361274
10spp	New		50 ms	0,0269132486	0,5514122248

Test made using Google Collaboratory: on pre rendered images

UNIVERSITÀ DEGLI STUDI DI UDINE

Denoiser in real-time application

UNIVERSITÄT KLAGENFURT

Image SPP	Is Denoiser ON	Image Example	Avg FPS	Min rendering time	Max rendering time
1spp	NO		2,802 FPS	98,0 ms	1148,7 ms
1spp	YES		2,538 FPS	123,0 ms	806,8 ms
5spp	NO		0,398 FPS	501,6 ms	6226,5 ms
5 spp	YES		0,388 FPS	546,3 ms	8032,9 ms

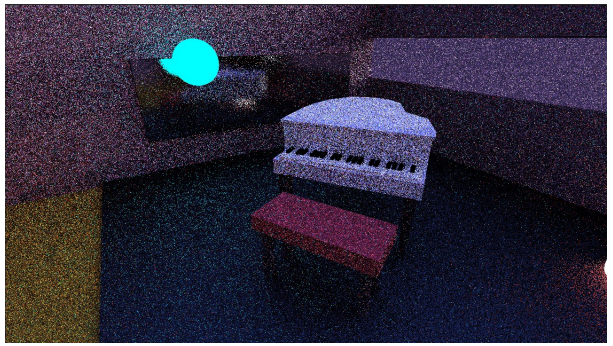
Test made on a PC equipped with Intel Core i5-5600k, Nvidia GTX 970 and 16Gb RAM

UNIVERSITÀ DEGLI STUDI DI UDINE

UNIVERSITÄT KLAGENFURT

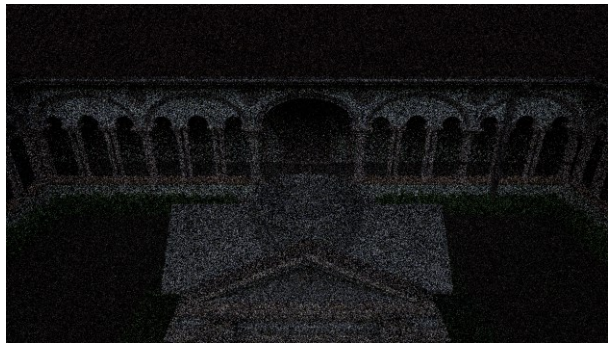
Conclusions

- Objectives reached:
  - Correct development of a real-time ray tracing application in Unity
  - Correct development of a Convolutional Neural Network for denoising of ray traced images
  - Correct use of the CNN in the real-time ray tracing application



1spp image (Project RP)

Denoised image



1spp image (Cycles Render)

Denoised image

	Real-time ray tracing application	Convolutional Neural Network	Real-time ray tracing application + CNN
Bonus	GPU parallel computation is faster than recursive one (CPU)	Denoiser quality compared to other denoisers like Cycles one	Low impact of CNN on rendering times (10,4% difference)
Malus	High rendering time (2,802 FPS)	Some chromatic aberrations	Ray tracing algorithm needs too computational power

CNN Vs CNN – extra comparison

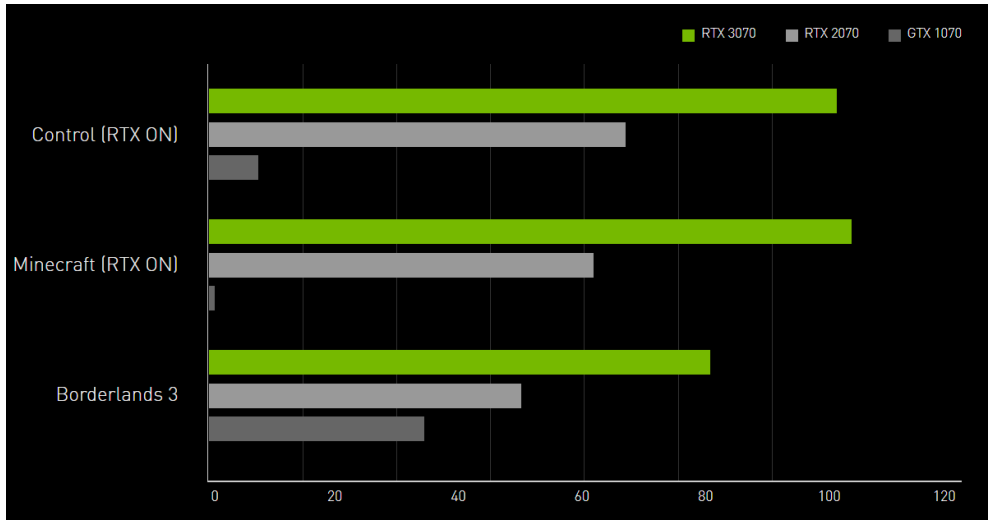
Denoised image via project denoiser

Denoised image via Optix (Arnold Renderer)

1spp image (Arnold Renderer)

Future work

- BVH implementation in ray tracing pipeline
- GPU improvement
  - RTX 30 series uses an architecture designed for real-time ray tracing
- Machine Learning for image resolution upscaling
  - Nvidia DLSS (Deep Learning Super Sample)



GPU computational comparison

DLSS comparison in Remedy's Control

UNIVERSITÀ DEGLI STUDI DI UDINE

UNIVERSITÄT KLAGENFURT

Thank you for your attention

Denoiser comparison

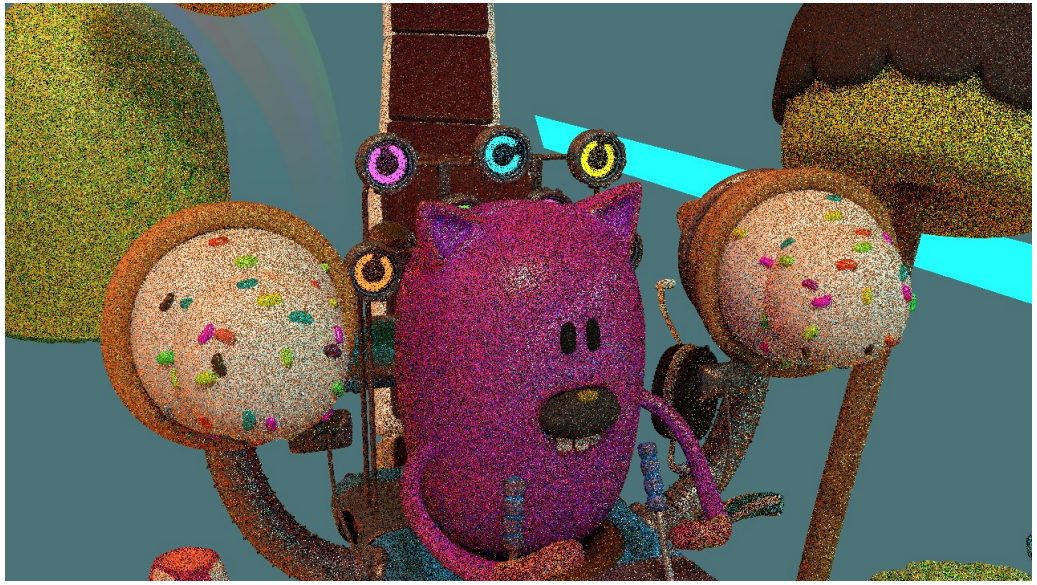
1 sample per pixel image

100 sample per pixel image + Cycles Denoiser

1 sample per pixel image + CNN Denoiser

1 sample per pixel image + Cycles Denoiser

Denoiser comparison



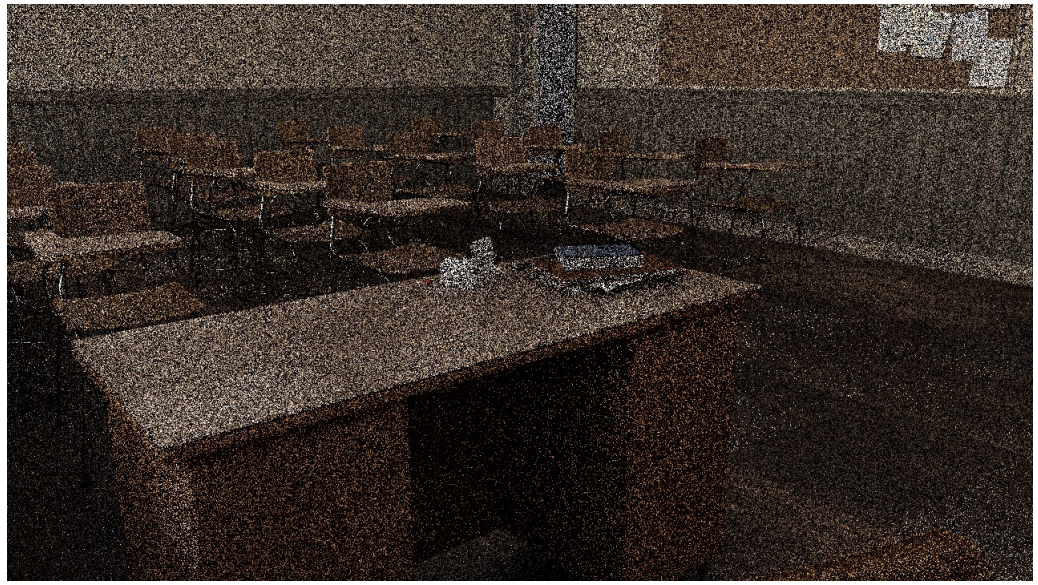
1 sample per pixel image

100 sample per pixel image + Cycles Denoiser

1 sample per pixel image + CNN Denoiser

1 sample per pixel image + Cycles Denoiser

Denoiser comparison



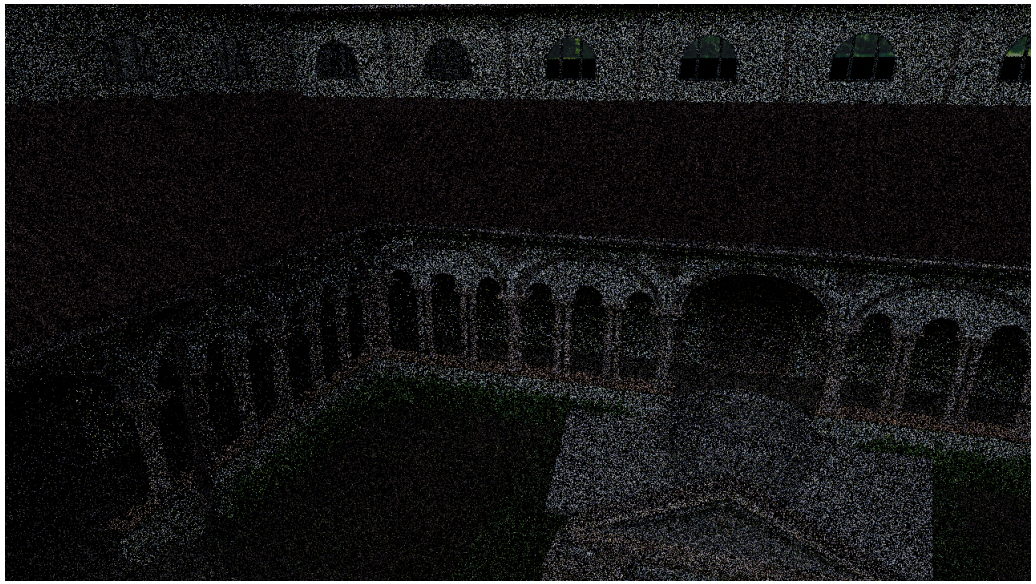
1 sample per pixel image

100 sample per pixel image + Cycles Denoiser

1 sample per pixel image + CNN Denoiser

1 sample per pixel image + Cycles Denoiser

Denoiser comparison



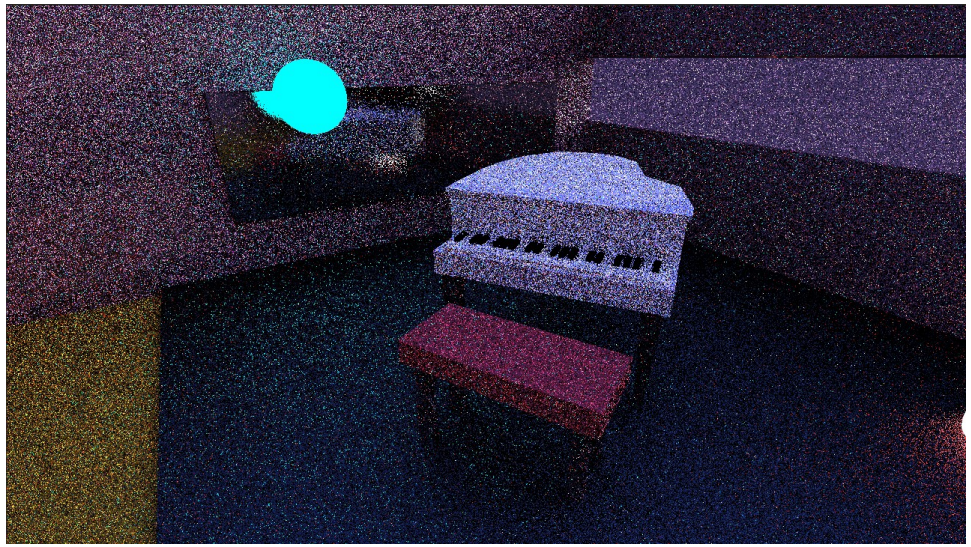
1 sample per pixel image

100 sample per pixel image + Cycles Denoiser

1 sample per pixel image + CNN Denoiser

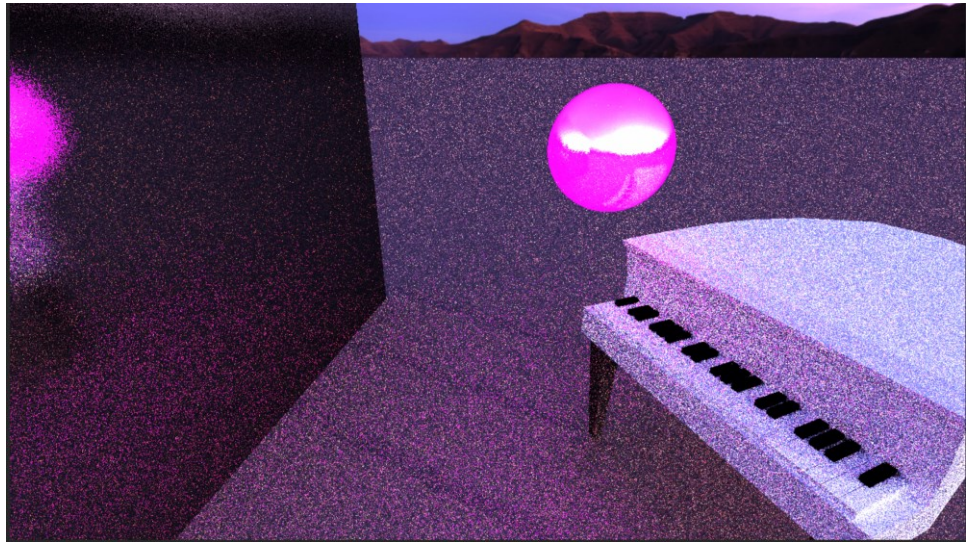
1 sample per pixel image + Cycles Denoiser

Ray tracing with and without denoiser

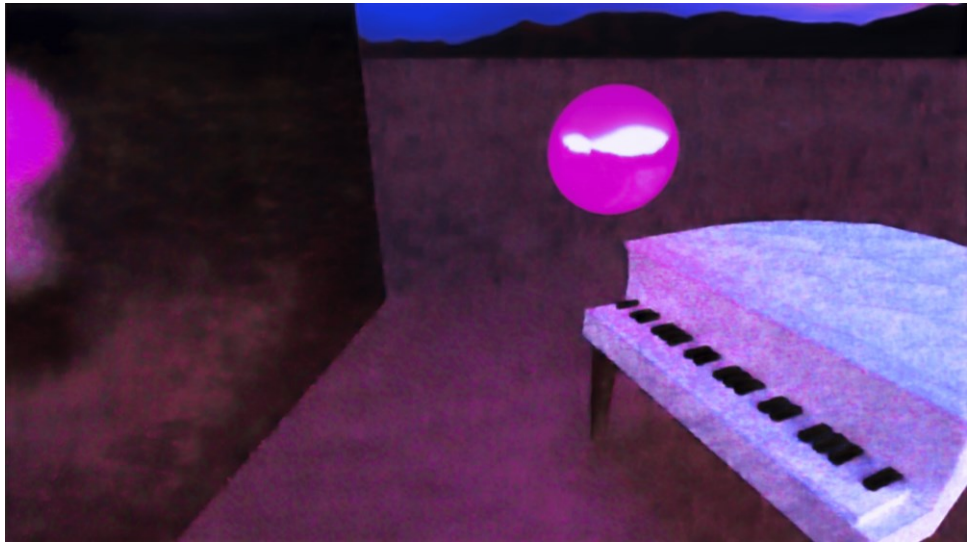


1 sample per pixel ray tracing (project)

1 sample per pixel ray tracing + CNN denoiser (project)



5 sample per pixel ray tracing (project)



5 sample per pixel ray tracing + CNN denoiser (project)

All rendering test made on a PC equipped with Intel Core i5-5600k, Nvidia GTX 970 and 16Gb RAM